

Game of Life - Német Bence UDTSJF

Készítette Doxygen 1.10.0

1. Hierarchikus mutató	1
1.1. Osztályhierarchia	1
2. Osztálymutató	3
2.1. Osztálylista	3
3. Osztályok dokumentációja	5
3.1. org.example.Cell osztályreferencia	5
3.1.1. Részletes leírás	7
3.1.2. Konstruktork és destruktorok dokumentációja	7
3.1.2.1. Cell() [1/2]	7
3.1.2.2. Cell() [2/2]	7
3.1.3. Tagfüggvények dokumentációja	7
3.1.3.1. borderOperation()	7
3.1.3.2. clearKill()	8
3.1.3.3. flip()	8
3.1.3.4. getDeadSince()	8
3.1.3.5. increaseDeadSince()	9
3.1.3.6. isAlive()	9
3.1.3.7. kill()	9
3.1.3.8. setAlive()	10
3.2. org.example.GameAreaPanel osztályreferencia	11
3.2.1. Részletes leírás	13
3.2.2. Konstruktork és destruktorok dokumentációja	13
3.2.2.1. GameAreaPanel()	13
3.2.3. Tagfüggvények dokumentációja	14
3.2.3.1. clearLivingSpace()	14
3.2.3.2. fadeColor()	14
3.2.3.3. getCellSize()	15
3.2.3.4. getPreferredSize()	15
3.2.3.5. isFadeOn()	16
3.2.3.6. isGridOn()	16
3.2.3.7. livingSpaceResize()	16
3.2.3.8. nextState()	17
3.2.3.9. paintComponent()	17
3.2.3.10. setColors()	18
3.2.3.11. setFadeOn()	18
3.2.3.12. setGridOn()	19
3.2.3.13. setLivingSpace()	19
3.3. org.example.GameControlPanel osztályreferencia	21
3.3.1. Részletes leírás	23
3.3.2. Konstruktork és destruktorok dokumentációja	23
3.3.2.1. GameControlPanel()	23

3.3.3.	Tagfüggvények dokumentációja	24
3.3.3.1.	load()	24
3.3.3.2.	parseRules()	25
3.3.3.3.	ruleSetter()	25
3.3.3.4.	run()	26
3.3.3.5.	save()	26
3.3.3.6.	startGame()	27
3.3.3.7.	stopGame()	27
3.4.	org.example.GameOfLife osztályreferencia	28
3.4.1.	Részletes leírás	30
3.4.2.	Konstruktorok és destruktorok dokumentációja	30
3.4.2.1.	GameOfLife()	30
3.4.3.	Tagfüggvények dokumentációja	30
3.4.3.1.	main()	30
3.5.	org.example.LivingSpace osztályreferencia	31
3.5.1.	Részletes leírás	33
3.5.2.	Konstruktorok és destruktorok dokumentációja	33
3.5.2.1.	LivingSpace()	33
3.5.3.	Tagfüggvények dokumentációja	33
3.5.3.1.	getAt()	33
3.5.3.2.	getCells()	34
3.5.3.3.	getSize()	34
3.5.3.4.	killAll()	35
3.5.3.5.	killAt()	35
3.5.3.6.	livingNeighbours()	35
3.5.3.7.	nextState()	36
3.5.3.8.	resize()	37
3.5.3.9.	reviveAt()	37
3.5.3.10.	setBornRule()	37
3.5.3.11.	setSurviveRule()	38

Tárgymutató	39
--------------------	-----------

1. fejezet

Hierarchikus mutató

1.1. Osztályhierarchia

Majdnem (de nem teljesen) betűrendbe szedett leszármazási lista:

JFrame	
org.example.GameOfLife	28
JPanel	
org.example.GameAreaPanel	11
org.example.GameControlPanel	21
Runnable	
org.example.GameControlPanel	21
Serializable	
org.example.Cell	5
org.example.LivingSpace	31

2. fejezet

Osztálymutató

2.1. Osztálylista

Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

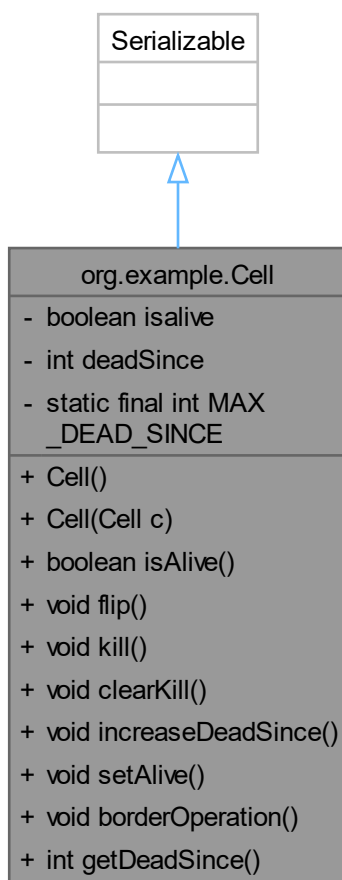
org.example.Cell	5
org.example.GameAreaPanel	11
org.example.GameControlPanel	21
org.example.GameOfLife	28
org.example.LivingSpace	31

3. fejezet

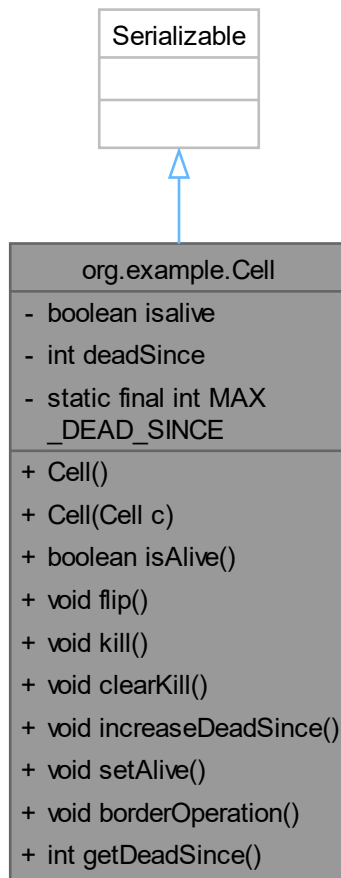
Osztályok dokumentációja

3.1. org.example.Cell osztályreferencia

Az org.example.Cell osztály származási diagramja:



Az org.example.Cell osztály együttműködési diagramja:



Publikus tagfüggvények

- [Cell \(\)](#)
- [Cell \(Cell c\)](#)
- [boolean isAlive \(\)](#)
- [void flip \(\)](#)
- [void kill \(\)](#)
- [void clearKill \(\)](#)
- [void increaseDeadSince \(\)](#)
- [void setAlive \(\)](#)
- [void borderOperation \(\)](#)
- [int getDeadSince \(\)](#)

Privát attribútumok

- [boolean isalive](#)
Cella állapota, false ha halott, true ha él.
- [int deadSince](#)
Cella halála óta eltelt generációk, a fade effekthez kell.

Statikus privát attribútumok

- `static final int MAX_DEAD_SINCE = 17`

Konstans, a deadSince max értéke.

3.1.1. Részletes leírás

Serializálható interfészt megvalósító Cell vagy sejt osztály

3.1.2. Konstruktorkok és destruktorkok dokumentációja

3.1.2.1. Cell() [1/2]

```
org.example.Cell.Cell ( )
```

Konstruktork, alapértelmezetten halottként inicializálja a cellát

3.1.2.2. Cell() [2/2]

```
org.example.Cell.Cell (
    Cell c )
```

Copy konstruktor

Paraméterek

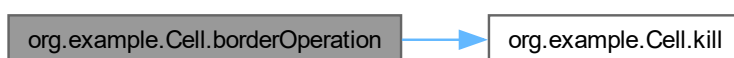
c	másolandó cella
---	-----------------

3.1.3. Tagfüggvények dokumentációja

3.1.3.1. borderOperation()

```
void org.example.Cell.borderOperation ( )
```

Szegélyen levő cellák kezelését segíti, megöli a cellát és a deadSince értékét maxra állítja A függvény hívási gráfja:



A függvény hívó gráfja:



3.1.3.2. clearKill()

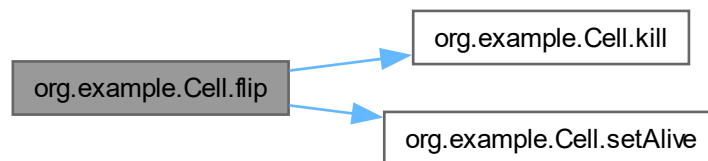
```
void org.example.Cell.clearKill ( )
```

Élettér clear-elésekor használt kill metódus, a deadSince értékét maxra állítja, a fade effekt miatt

3.1.3.3. flip()

```
void org.example.Cell.flip ( )
```

A jelenlegi állapot ellentétére váltja a cellát. A függvény hívási gráfja:



3.1.3.4. getDeadSince()

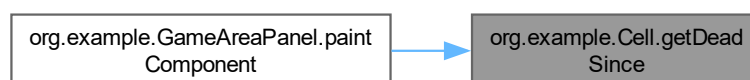
```
int org.example.Cell.getDeadSince ( )
```

Visszaadja a deadSince értékét

Visszatérési érték

deadSince értéke

A függvény hívó gráfja:



3.1.3.5. increaseDeadSince()

```
void org.example.Cell.increaseDeadSince ( )
```

Növeli a halott cella deadSince értékét, legfeljebb ötig A függvény hívó gráfja:



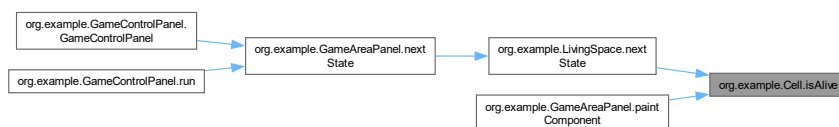
3.1.3.6. isAlive()

```
boolean org.example.Cell.isAlive ( )
```

Visszatérési érték

a cella állapota

A függvény hívó gráfja:



3.1.3.7. kill()

```
void org.example.Cell.kill ( )
```

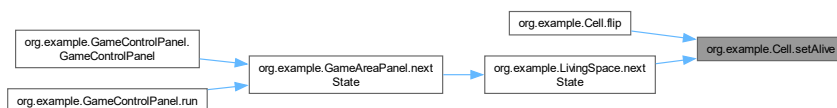
Megöli a cellát és egyre állítja a deadSince értékét A függvény hívó gráfja:



3.1.3.8. setAlive()

```
void org.example.Cell.setAlive ( )
```

Feléleszti a cellát, a deadSince értékét nullára állítja A függvény hívó gráfja:

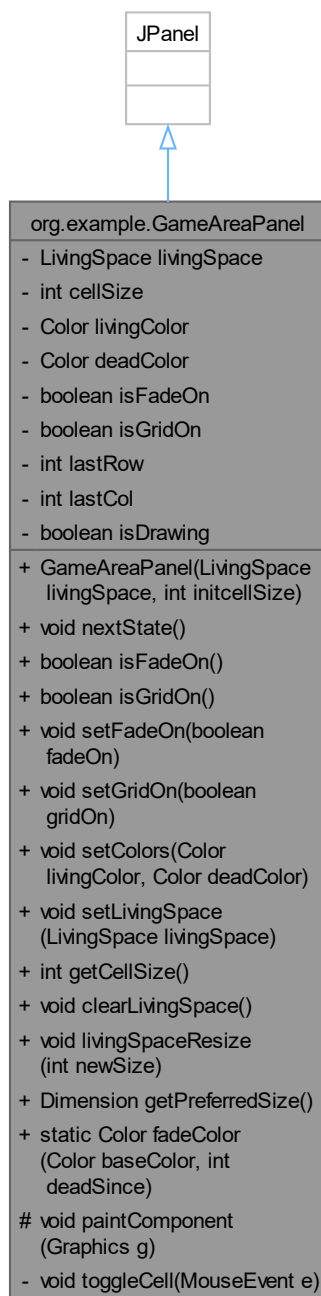


Ez a dokumentáció az osztályról a következő fájl alapján készült:

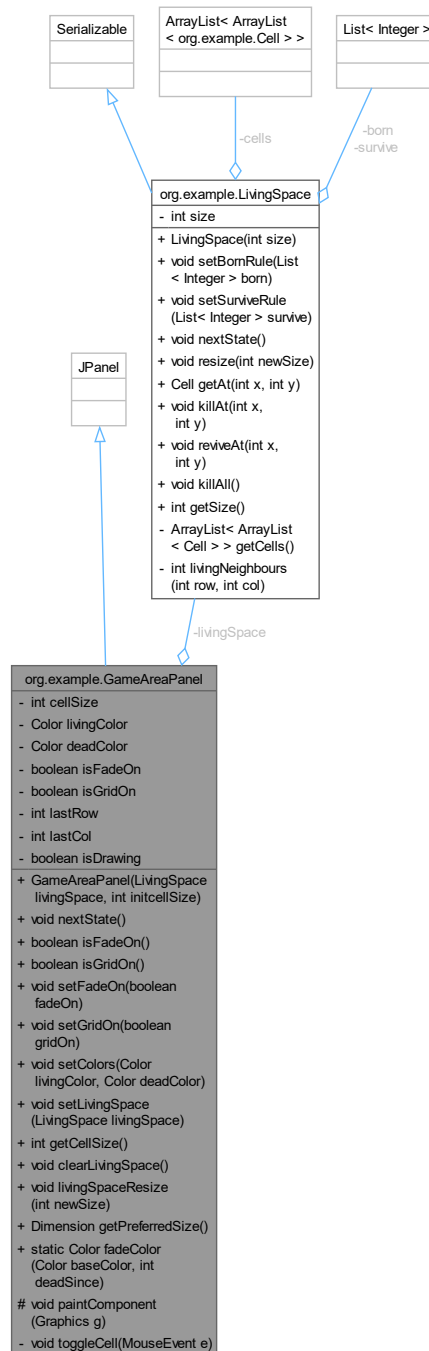
- `src/main/java/org/example/Cell.java`

3.2. org.example.GameAreaPanel osztályreferencia

Az org.example.GameAreaPanel osztály származási diagramja:



Az org.example.GameAreaPanel osztály együttműködési diagramja:



Publikus tagfüggvények

- **GameAreaPanel** (**LivingSpace** livingSpace, int initcellSize)
- void **nextState** ()
- boolean **isFadeOn** ()
- boolean **isGridOn** ()
- void **setFadeOn** (boolean fadeOn)

- void `setGridOn` (boolean `gridOn`)
- void `setColors` (Color `livingColor`, Color `deadColor`)
- void `setLivingSpace` (LivingSpace `livingSpace`)
- int `getCellSize` ()
- void `clearLivingSpace` ()
- void `livingSpaceResize` (int `newSize`)
- Dimension `getPreferredSize` ()

Statikus publikus tagfüggvények

- static Color `fadeColor` (Color `baseColor`, int `deadSince`)

Védett tagfüggvények

- void `paintComponent` (Graphics `g`)

Privát tagfüggvények

- void `toggleCell` (MouseEvent `e`)

Privát attribútumok

- LivingSpace `livingSpace`
A panelhoz tartozó élettér példány.
- int `cellSize`
Egy cella pixelben megadott mérete.
- Color `livingColor`
Élő cellák színe.
- Color `deadColor`
Halott cellák színe.
- boolean `isFadeOn` = false
Halott cellák fokozatos elhalványozásának engedélyezése.
- boolean `isGridOn` = true
Rács kirajzolásának engedélyezése.
- int `lastRow` = -1
- int `lastCol` = -1
Utoljára rajzolt sor és oszlop.
- boolean `isDrawing` = false

3.2.1. Részletes leírás

JPanelből származtatott osztály, az élettér grafikus megjelenítése a szerepe

3.2.2. Konstruktorok és destruktorok dokumentációja

3.2.2.1. GameAreaPanel()

```
org.example.GameAreaPanel.GameAreaPanel (
    LivingSpace livingSpace,
    int initcellSize )
```

GameAreaPanel konstruktora. Beállítja az eventlistenereket az egérműveletekhez (Kattintással rajzolás, görgetéssel zoom)

Paraméterek

<i>livingSpace</i>	A panelhez tartozó élettér példány
<i>initcellSize</i>	A kezdeti cellaméret (zoom miatt később változhat)

3.2.3. Tagfüggvények dokumentációja

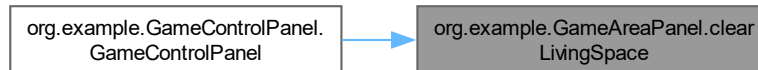
3.2.3.1. clearLivingSpace()

```
void org.example.GameAreaPanel.clearLivingSpace ( )
```

Megöli a panelhez tartozó élettér összes celláját A függvény hívási gráfja:



A függvény hívó gráfja:



3.2.3.2. fadeColor()

```
static Color org.example.GameAreaPanel.fadeColor (
    Color baseColor,
    int deadSince ) [static]
```

Egy szín elhalványodását megoldó segédfüggvény

Paraméterek

<i>baseColor</i>	módosítani kívánt szín
<i>deadSince</i>	fade mértéke, 0 és 5 között

Visszatérési érték

kívánt mértékben elhalványított szín

A függvény hívó gráfja:

**3.2.3.3. getCellSize()**

```
int org.example.GameAreaPanel.getCellSize ( )
```

Visszaadja az aktuális cellaméretet

Visszatérési érték

cellaméret

3.2.3.4. getPreferredSize()

```
Dimension org.example.GameAreaPanel.getPreferredSize ( )
```

Beállítja a panel méretét az élettér és a cellaméret szerint

Visszatérési érték

Dimension példány a számított méretekkel

A függvény hívási gráfja:



3.2.3.5. isFadeOn()

```
boolean org.example.GameAreaPanel.isFadeOn ( )
```

Visszatérési érték

boolean, be van-e kapcsolva az elhalványodás effekt

3.2.3.6. isGridOn()

```
boolean org.example.GameAreaPanel.isGridOn ( )
```

Visszatérési érték

boolean, be van-e kapcsolva a rács kirajzolása

3.2.3.7. livingSpaceResize()

```
void org.example.GameAreaPanel.livingSpaceResize (
    int newSize )
```

Átméretezi a panelhez tartozó életteret

Paraméterek

<i>newSize</i>	új méret
----------------	----------

A függvény hívási gráfja:



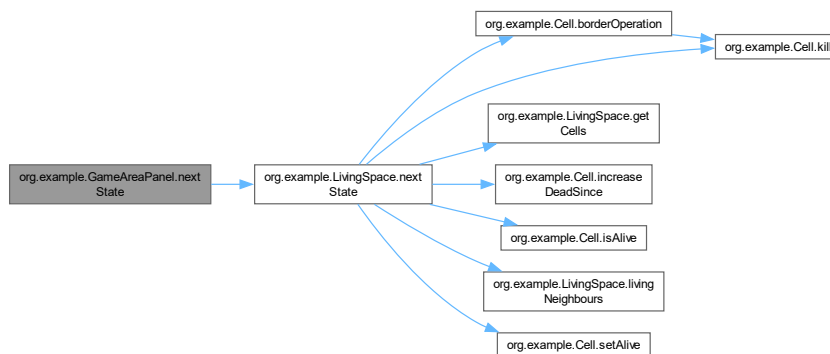
A függvény hívó gráfja:



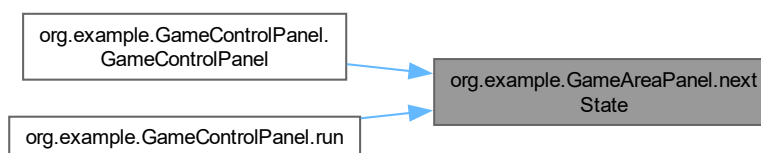
3.2.3.8. nextState()

```
void org.example.GameAreaPanel.nextState ( )
```

Lépteti a panelhez tartozó életteret, majd frissíti a panel megjelenését. A függvény hívási gráfja:



A függvény hívó gráfja:



3.2.3.9. paintComponent()

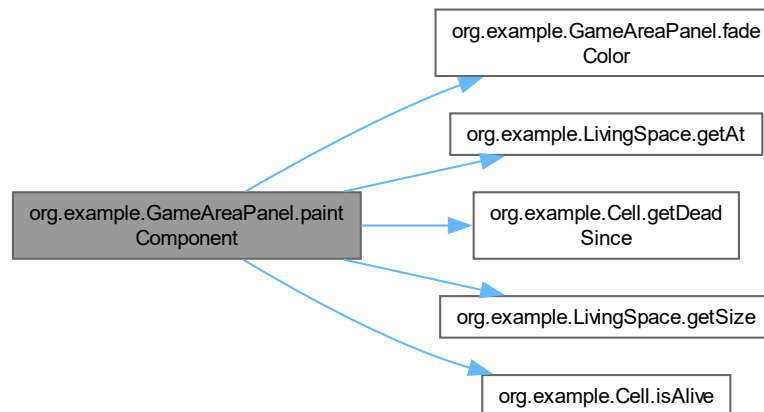
```
void org.example.GameAreaPanel.paintComponent (
    Graphics g ) [protected]
```

Kirajzolja az élettér jelenlegi állapotát a beállított grid és fade változókat figyelembe véve

Paraméterek

<i>g</i>	egy Graphics példány ami a rajzolást végzi
----------	--

A függvény hívási gráfja:



3.2.3.10. setColors()

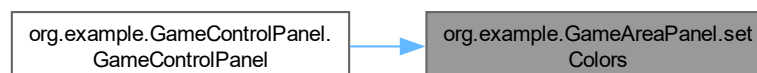
```
void org.example.GameAreaPanel.setColors (
    Color livingColor,
    Color deadColor )
```

Beállítja a cellák színeit

Paraméterek

<i>livingColor</i>	élő cellák színe
<i>deadColor</i>	halott cellák színe

A függvény hívó gráfja:



3.2.3.11. setFadeOn()

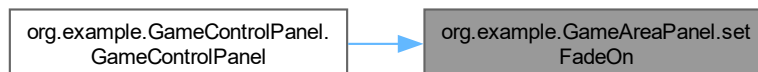
```
void org.example.GameAreaPanel.setFadeOn (
    boolean fadeOn )
```

Beállítja az elhalványodás effekt engedélyezését

Paraméterek

<i>fadeOn</i>	effekt állapota
---------------	-----------------

A függvény hívó gráfja:

**3.2.3.12. setGridOn()**

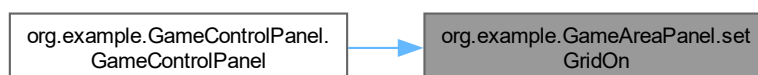
```
void org.example.GameAreaPanel.setGridOn (
    boolean gridOn )
```

Beállítja a rács kirajzolásának engedélyezését

Paraméterek

<i>gridOn</i>	rácsrajzolás állapota
---------------	-----------------------

A függvény hívó gráfja:

**3.2.3.13. setLivingSpace()**

```
void org.example.GameAreaPanel.setLivingSpace (
    LivingSpace livingSpace )
```

Frissíti a referenciát a paraméterként megadott új élettérre, és frissíti a hozzá tartozó eventlistenereket

Paraméterek

<i>livingSpace</i>	új élettér referenciája
--------------------	-------------------------

A függvény hívó gráfja:

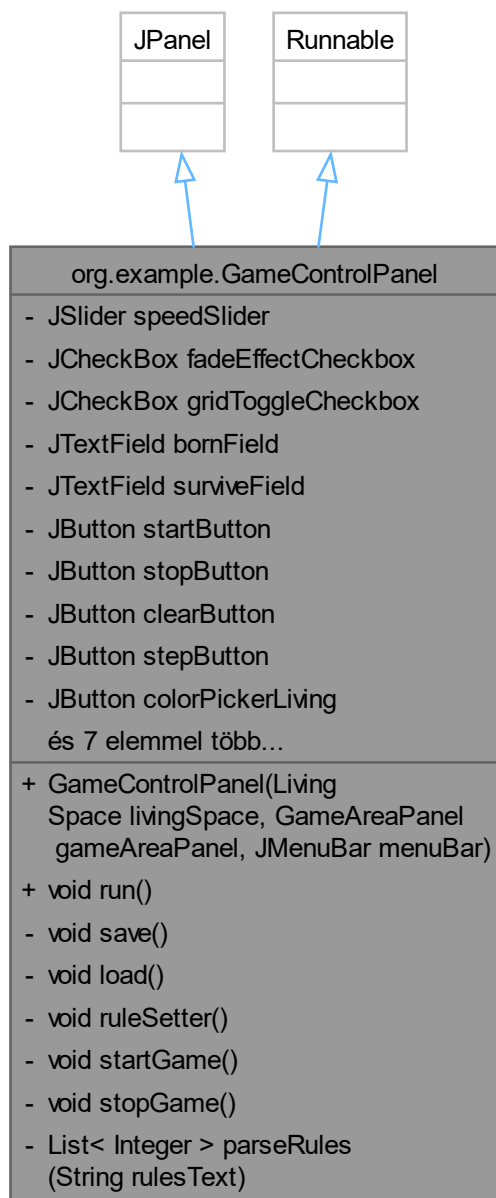


Ez a dokumentáció az osztályról a következő fájl alapján készült:

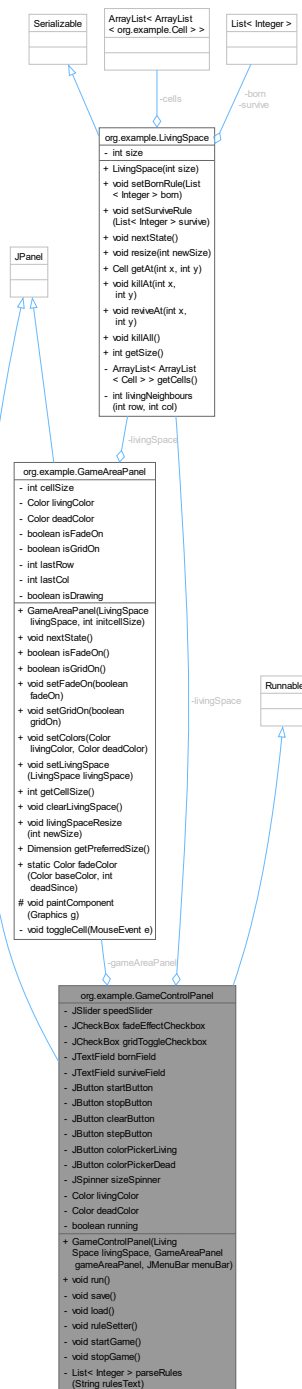
- src/main/java/org/example/GameAreaPanel.java

3.3. org.example.GameControlPanel osztályreferencia

Az org.example.GameControlPanel osztály származási diagramja:



Az org.example.GameControlPanel osztály együttműködési diagramja:



Publikus tagfüggvények

- **GameControlPanel** (**LivingSpace** livingSpace, **GameAreaPanel** gameAreaPanel, JMenuBar menuBar)
- void **run** ()

Privát tagfüggvények

- void **save** ()

- void `load` ()
- void `ruleSetter` ()
- void `startGame` ()
- void `stopGame` ()
- List< Integer > `parseRules` (String rulesText) throws NumberFormatException

Privát attribútumok

- JSlider `speedSlider`
- JCheckBox `fadeEffectCheckbox`
- JCheckBox `gridToggleCheckbox`
- JTextField `bornField`
- JTextField `surviveField`
- JButton `startButton`
- JButton `stopButton`
- JButton `clearButton`
- JButton `stepButton`
- JButton `colorPickerLiving`
- JButton `colorPickerDead`
- `LivingSpace` `livingSpace`
- JSpinner `sizeSpinner`
- Color `livingColor` = Color.blue
Alapértelmezetten kékek az élő cellák.
- Color `deadColor` = Color.white
Alapértelmezetten fehérek a halott cellák.
- `GameAreaPanel` `gameAreaPanel`
A vezérelt GameAreaPanel példány.
- boolean `running` = false

3.3.1. Részletes leírás

A játék vezérlését végző menü osztálya, JPanelből leszármazva és a Runnable interfészt implementálja

3.3.2. Konstruktorkok és destruktorok dokumentációja

3.3.2.1. GameControlPanel()

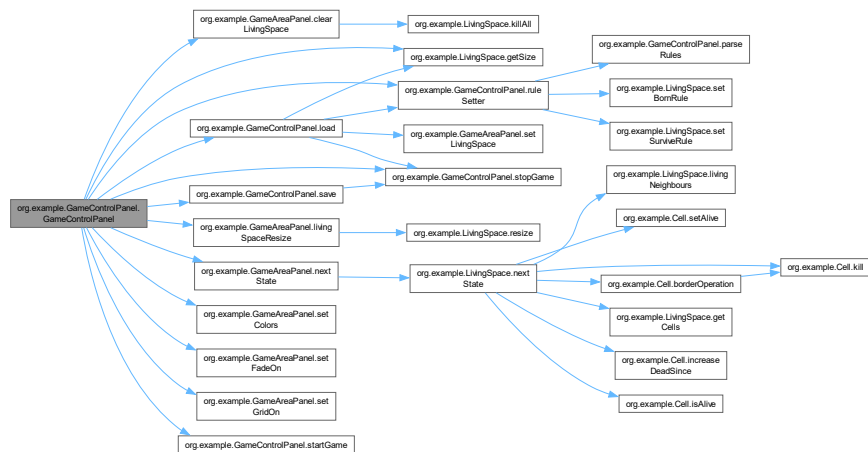
```
org.example.GameControlPanel.GameControlPanel (
    LivingSpace livingSpace,
    GameAreaPanel gameAreaPanel,
    JMenuBar menuBar )
```

Konstruktork, beállítja a GUI elemeit, a hozzájuk tartozó eventlistenereket, valamint a paraméterben átvett GameAreaPanel és LivingSpace példányokat

Paraméterek

<code>livingSpace</code>	vezérelt élettér
<code>gameAreaPanel</code>	vezérelt GameAreaPanel amihez az élettér is tartozik

A függvény hívási gráfja:

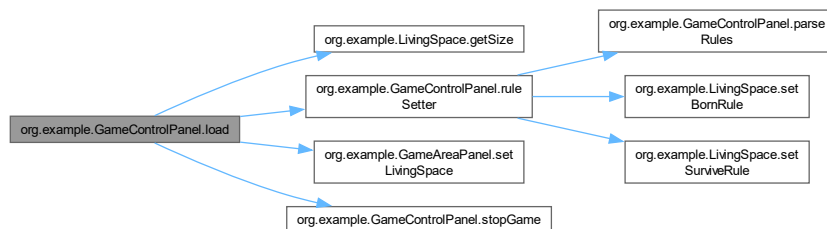


3.3.3. Tagfüggvények dokumentációja

3.3.3.1. load()

```
void org.example.GameControlPanel.load ( ) [private]
```

Megállítja a szimulációt, egy felugró ablakban kiválasztott .dat kiterjesztésű fájlból betölt egy abba szerializálva mentett élettér példányt, majd frissíti a szükséges referenciákat. A függvény hívási gráfja:



A függvény hívó gráfja:



3.3.3.2. parseRules()

```
List< Integer > org.example.GameControlPanel.parseRules (
    String rulesText ) throws NumberFormatException [private]
```

Segédmetódus a szabálymező parseolásához. Szigorúan csak "0,1,2,3..." formátumban fogadja el a szabályokat. Ha parseoláskor hiba lép fel, az alapéltelmezett Conway-féle szabályokat állítja be (B3/S23).

Paraméterek

<i>rulesText</i>	String ami a szabályokat tartalmazza
------------------	--------------------------------------

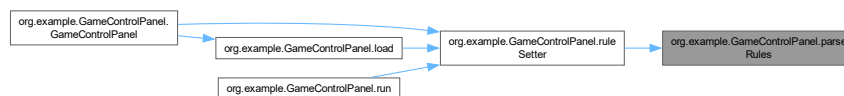
Visszatérési érték

List a rulesTextből kiválogatott Integerekből

Kivételek

<i>NumberFormatException</i>	ha sikertelen a parseolás
------------------------------	---------------------------

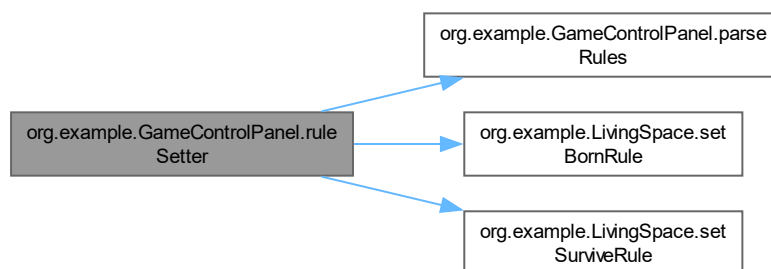
A függvény hívó gráfja:



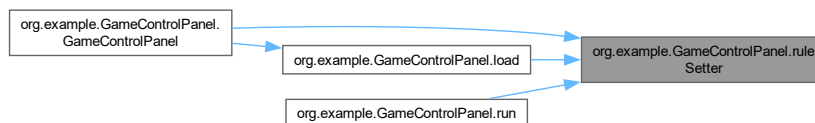
3.3.3.3. ruleSetter()

```
void org.example.GameControlPanel.ruleSetter ( ) [private]
```

Beállítja az szimuláció szabályait a két vonatkozó szövegmező tartalma szerint. A függvény hívási gráfja:



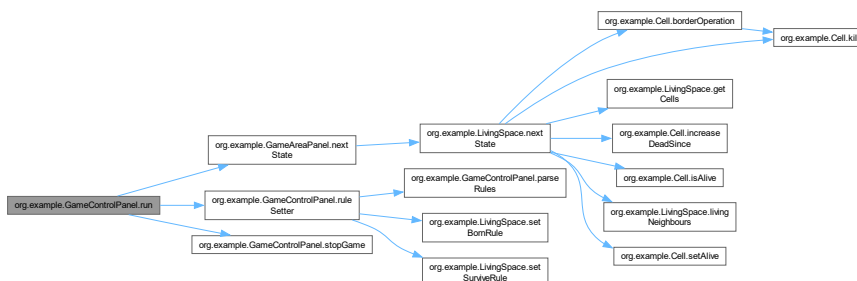
A függvény hívó gráfja:



3.3.3.4. run()

```
void org.example.GameControlPanel.run ( )
```

Szimuláció folyamatos futását implementáló függvény. Sebessége a speedSlider értékének függvénye, amit folyamatosan lekér A függvény hívási gráfja:



3.3.3.5. save()

```
void org.example.GameControlPanel.save ( ) [private]
```

Megállítja a szimulációt és egy felugró ablakban kiválasztott fájlba elmenti annak jelenlegi állapotát, szerializálással. A függvény hívási gráfja:



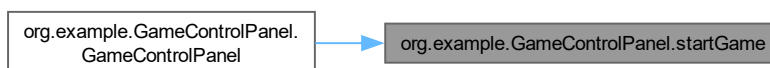
A függvény hívó gráfja:



3.3.3.6. startGame()

```
void org.example.GameControlPanel.startGame ( ) [private]
```

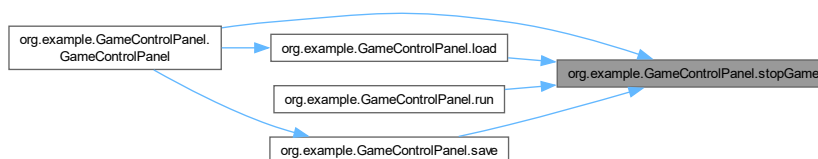
Elindít egy szálat a szimuláció folyamatos futásához A függvény hívó gráfja:



3.3.3.7. stopGame()

```
void org.example.GameControlPanel.stopGame ( ) [private]
```

Megállítja a szimulációt, a hozzá tartozó szálat interruptolja A függvény hívó gráfja:

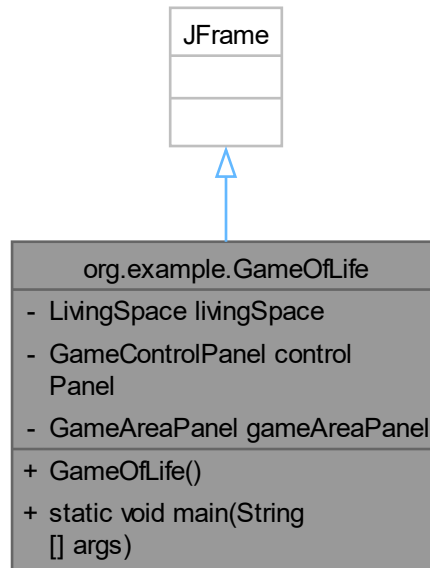


Ez a dokumentáció az osztályról a következő fájl alapján készült:

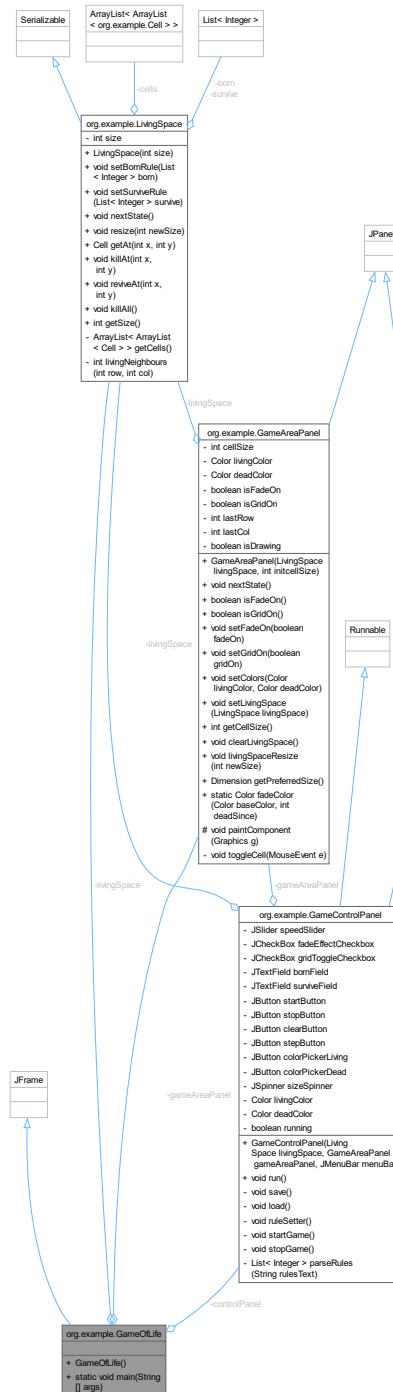
- `src/main/java/org/example/GameControlPanel.java`

3.4. org.example.GameOfLife osztályreferencia

Az org.example.GameOfLife osztály származási diagramja:



Az org.example.GameOfLife osztály együttműködési diagramja:



Publikus tagfüggvények

- [GameOfLife](#) ()

Statikus publikus tagfüggvények

- static void [main](#) (String[] args)

Privát attribútumok

- [LivingSpace](#) **livingSpace**
A futáshoz tartozó élettér.
- [GameControlPanel](#) **controlPanel**
A futáshoz tartozó GameControlPanel.
- [GameAreaPanel](#) **gameAreaPanel**
A futáshoz tartozó GameAreaPanel.

3.4.1. Részletes leírás

A program fő osztálya

3.4.2. Konstruktorkok és destruktorkok dokumentációja

3.4.2.1. GameOfLife()

```
org.example.GameOfLife.GameOfLife ( )
```

Konstruktork, beállít pár GUI elemet, egy alapértelmezetten 50x50-es életteret, valamint a livingSpace, controlPanel gameAreaPanel mezőket

3.4.3. Tagfüggvények dokumentációja

3.4.3.1. main()

```
static void org.example.GameOfLife.main (
    String[] args ) [static]
```

Main metódus, indítja a programot.

Paraméterek

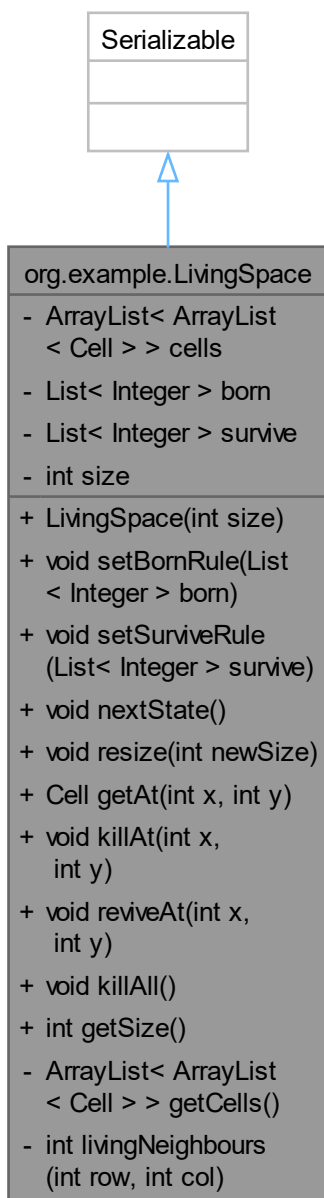
<i>args</i>	
-------------	--

Ez a dokumentáció az osztályról a következő fájl alapján készült:

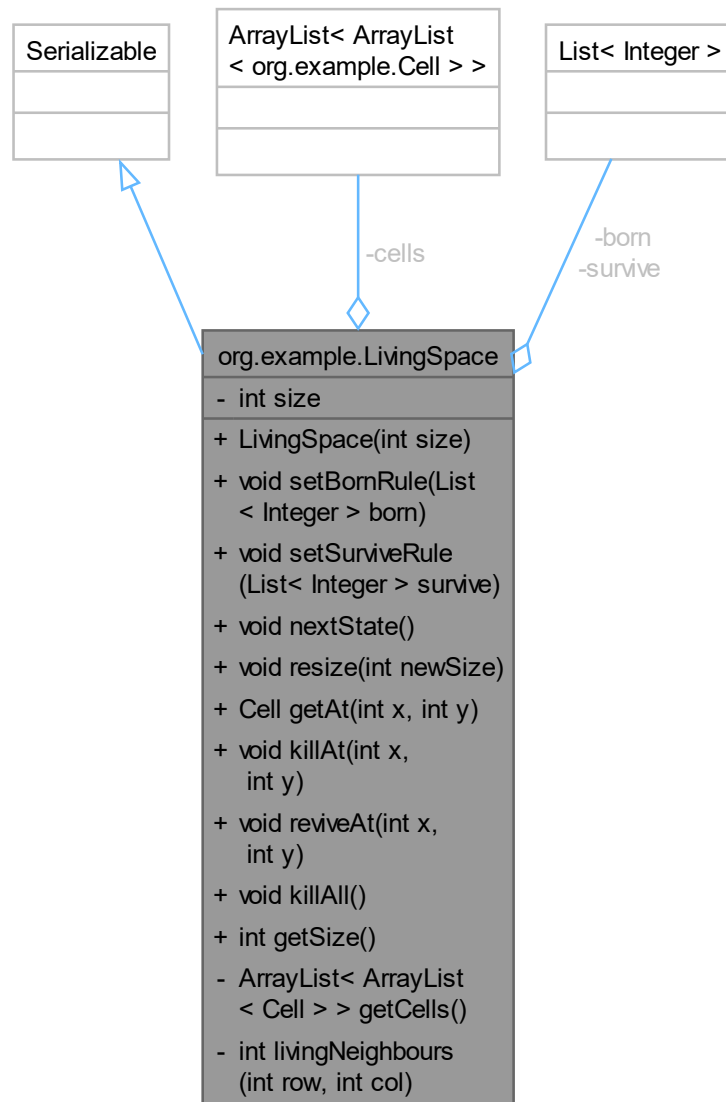
- src/main/java/org/example/GameOfLife.java

3.5. org.example.LivingSpace osztályreferencia

Az org.example.LivingSpace osztály származási diagramja:



Az org.example.LivingSpace osztály együttműködési diagramja:



Publikus tagfüggvények

- `LivingSpace` (int `size`)
- void `setBornRule` (List< Integer > `born`)
- void `setSurviveRule` (List< Integer > `survive`)
- void `nextState` ()
- void `resize` (int `newSize`)
- `Cell` `getAt` (int `x`, int `y`)
- void `killAt` (int `x`, int `y`)
- void `reviveAt` (int `x`, int `y`)
- void `killAll` ()
- int `getSize` ()

Privát tagfüggvények

- ArrayList< ArrayList< Cell > > **getCells** ()
- int **livingNeighbours** (int row, int col)

Privát attribútumok

- ArrayList< ArrayList< Cell > > **cells**
Cellákból álló kétdimenziós ArrayList.
- List< Integer > **born**
Születési szabályok.
- List< Integer > **survive**
Túlélési szabályok.
- int **size**
Élettér mérete.

3.5.1. Részletes leírás

LivingSpace vagy magyarul élettér osztály. Tartalmazza a sejtautomata mátrixát, annak méretét és celláit, a szimuláció szabályait. A Serializable interfészt megvalósítja, a fájlba menthetőség érdekében

3.5.2. Konstruktorkok és destruktorok dokumentációja**3.5.2.1. LivingSpace()**

```
org.example.LivingSpace.LivingSpace (
    int size )
```

LivingSpace konstruktora. Minden cellát halottként inicializál

Paraméterek

<i>size</i>	Az élettér mérete. Mivel mindig négyzetes, csak egy paraméter
-------------	---

3.5.3. Tagfüggvények dokumentációja**3.5.3.1. getAt()**

```
Cell org.example.LivingSpace.getAt (
    int x,
    int y )
```

Visszaad egy koordinátaival megadott cellát.

Paraméterek

<i>x</i>	X koordináta
<i>y</i>	Y koordináta

Visszatérési érték

Cella a megadott helyen

A függvény hívó gráfja:

**3.5.3.2. getCells()**

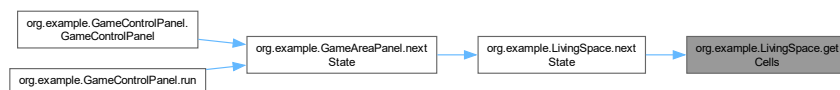
```
ArrayList< ArrayList< Cell > > org.example.LivingSpace.getCells ( ) [private]
```

Visszaad egy másolatot az élettér celláiról

Visszatérési érték

Kétdimenziós ArrayList az élettér aktuális állapotát reprezentálva

A függvény hívó gráfja:

**3.5.3.3. getSize()**

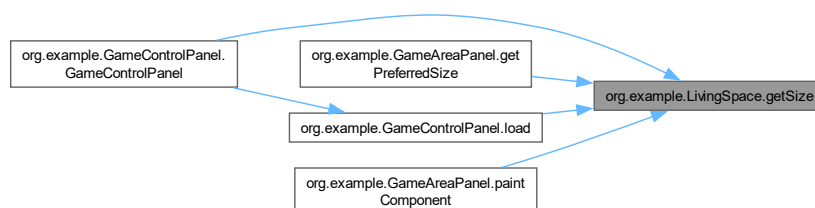
```
int org.example.LivingSpace.getSize ( )
```

Visszaadja az élettér méretét

Visszatérési érték

Az élettér mérete

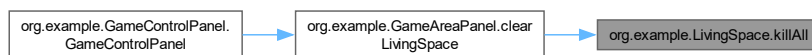
A függvény hívó gráfja:



3.5.3.4. killAll()

```
void org.example.LivingSpace.killAll ( )
```

Megöli az élettér összes celláját A függvény hívó gráfja:



3.5.3.5. killAt()

```
void org.example.LivingSpace.killAt (
    int x,
    int y )
```

Megöli a koordinátaival megadott cellát

Paraméterek

<i>x</i>	X koordináta
<i>y</i>	Y koordináta

3.5.3.6. livingNeighbours()

```
int org.example.LivingSpace.livingNeighbours (
    int row,
    int col ) [private]
```

Összeszámolja egy koordinátaival megadott cella élő szomszédait

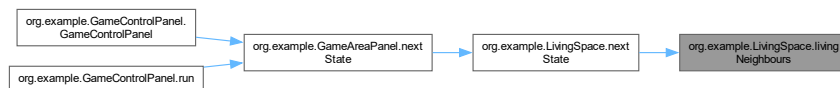
Paraméterek

<i>row</i>	Cella Y koordinátája
<i>col</i>	Cella X koordinátája

Visszatérési érték

Élő szomszédok száma

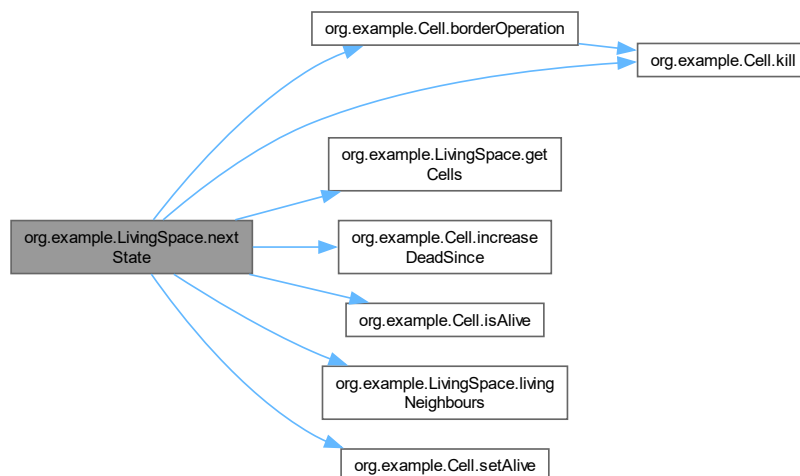
A függvény hívó gráfja:



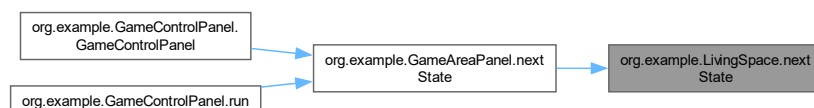
3.5.3.7. nextState()

```
void org.example.LivingSpace.nextState ( )
```

Következő állapotba lépteti az életteret az aktuális szabályok szerint A szegélyen levő cellákat minden körben megöli, ezzel adva határt az élettérnek A függvény hívási gráfja:



A függvény hívó gráfja:



3.5.3.8. resize()

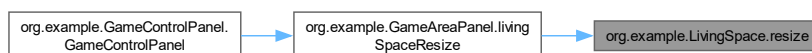
```
void org.example.LivingSpace.resize (
    int newSize )
```

Átméretezi az életteret. A bal felső sarok felé zsugorítja vagy a jobb alsó sarok felé nyújtja. Átmásolja a jelenlegi állapotnak azt a részét, ami belefér az új élettérbe

Paraméterek

<i>newSize</i>	Az élettér új mérete
----------------	----------------------

A függvény hívó gráfja:



3.5.3.9. reviveAt()

```
void org.example.LivingSpace.reviveAt (
    int x,
    int y )
```

Feléleszti a koordinátaival megadott cellát

Paraméterek

<i>x</i>	X koordináta
<i>y</i>	Y koordináta

3.5.3.10. setBornRule()

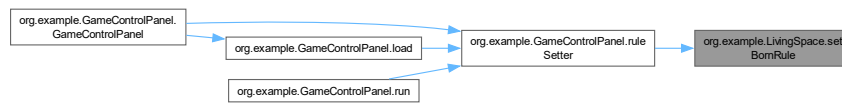
```
void org.example.LivingSpace.setBornRule (
    List< Integer > born )
```

Beállítja a születési szabályokat

Paraméterek

<i>born</i>	Egy egész számokból álló lista.
-------------	---------------------------------

A függvény hívó gráfja:



3.5.3.11. setSurviveRule()

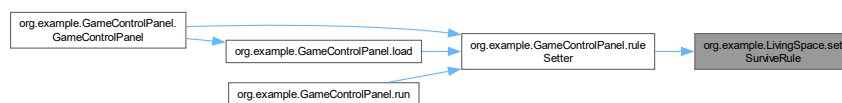
```
void org.example.LivingSpace.setSurviveRule (
    List< Integer > survive )
```

Beállítja a túlélési szabályokat

Paraméterek

<i>survive</i>	Egy egész számokból álló lista
----------------	--------------------------------

A függvény hívó gráfja:



Ez a dokumentáció az osztályról a következő fájl alapján készült:

- `src/main/java/org/example/LivingSpace.java`

Tárgymutató

borderOperation	org.example.Cell, 7
Cell	org.example.Cell, 7
clearKill	org.example.Cell, 8
clearLivingSpace	org.example.GameAreaPanel, 14
fadeColor	org.example.GameAreaPanel, 14
flip	org.example.Cell, 8
GameAreaPanel	org.example.GameAreaPanel, 13
GameControlPanel	org.example.GameControlPanel, 23
GameOfLife	org.example.GameOfLife, 30
getAt	org.example.LivingSpace, 33
getCells	org.example.LivingSpace, 34
getCellSize	org.example.GameAreaPanel, 15
getDeadSince	org.example.Cell, 8
getPreferredSize	org.example.GameAreaPanel, 15
getSize	org.example.LivingSpace, 34
increaseDeadSince	org.example.Cell, 8
isAlive	org.example.Cell, 9
isFadeOn	org.example.GameAreaPanel, 15
isGridOn	org.example.GameAreaPanel, 16
kill	org.example.Cell, 9
killAll	org.example.LivingSpace, 34
killAt	org.example.LivingSpace, 35
livingNeighbours	org.example.LivingSpace, 35
LivingSpace	org.example.LivingSpace, 33
livingSpaceResize	org.example.GameAreaPanel, 16
load	org.example.GameControlPanel, 24
main	org.example.GameOfLife, 30
nextState	org.example.GameAreaPanel, 16 org.example.LivingSpace, 36
org.example.Cell	5 borderOperation, 7 Cell, 7 clearKill, 8 flip, 8 getDeadSince, 8 increaseDeadSince, 8 isAlive, 9 kill, 9 setAlive, 9
org.example.GameAreaPanel	11 clearLivingSpace, 14 fadeColor, 14 GameAreaPanel, 13 getCellSize, 15 getPreferredSize, 15 isFadeOn, 15 isGridOn, 16 livingSpaceResize, 16 nextState, 16 paintComponent, 17 setColors, 18 setFadeOn, 18 setGridOn, 19 setLivingSpace, 19
org.example.GameControlPanel	21 GameControlPanel, 23 load, 24 parseRules, 24 ruleSetter, 25 run, 26 save, 26 startGame, 26 stopGame, 27
org.example.GameOfLife	28

- GameOfLife, [30](#)
 - main, [30](#)
- org.example.LivingSpace, [31](#)
 - getAt, [33](#)
 - getCells, [34](#)
 - getSize, [34](#)
 - killAll, [34](#)
 - killAt, [35](#)
 - livingNeighbours, [35](#)
 - LivingSpace, [33](#)
 - nextState, [36](#)
 - resize, [36](#)
 - reviveAt, [37](#)
 - setBornRule, [37](#)
 - setSurviveRule, [38](#)
- paintComponent
 - org.example.GameAreaPanel, [17](#)
- parseRules
 - org.example.GameControlPanel, [24](#)
- resize
 - org.example.LivingSpace, [36](#)
- reviveAt
 - org.example.LivingSpace, [37](#)
- ruleSetter
 - org.example.GameControlPanel, [25](#)
- run
 - org.example.GameControlPanel, [26](#)
- save
 - org.example.GameControlPanel, [26](#)
- setAlive
 - org.example.Cell, [9](#)
- setBornRule
 - org.example.LivingSpace, [37](#)
- setColors
 - org.example.GameAreaPanel, [18](#)
- setFadeOn
 - org.example.GameAreaPanel, [18](#)
- setGridOn
 - org.example.GameAreaPanel, [19](#)
- setLivingSpace
 - org.example.GameAreaPanel, [19](#)
- setSurviveRule
 - org.example.LivingSpace, [38](#)
- startGame
 - org.example.GameControlPanel, [26](#)
- stopGame
 - org.example.GameControlPanel, [27](#)